

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-149414

(43)Date of publication of application : 02.06.1999

(51)Int.Cl. G06F 12/14
G09C 1/00
H04L 9/08
H04L 9/32

(21)Application number : 10-199966

(71)Applicant : HITACHI SOFTWARE ENG CO LTD

(22)Date of filing : 15.07.1998

(72)Inventor : MIYAZAKI HIROSHI
SAMEJIMA YOSHIKI

(30)Priority

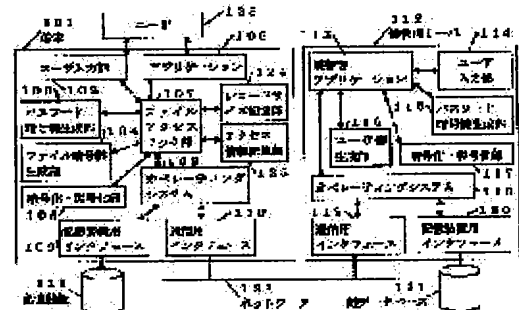
Priority number : 09247379 Priority date : 12.09.1997 Priority country : JP

(54) METHOD FOR PRESERVING DATA, ITS SYSTEM AND STORAGE MEDIUM FOR DATA PRESERVATION PROCESSING

(57)Abstract:

PROBLEM TO BE SOLVED: To strongly protect encipherment data by obtaining a cryptographic key required for ciphering and deciphering data from a key management computer being different from a computer to which a storage device for preserving encipher data is connected.

SOLUTION: The file cryptographic key generating part 104 of a terminal 101 generates the cryptographic key to be used for ciphering a file at random, which is preserved in the storage device, and a ciphering and deciphering part 105 actually ciphers and decipheres the file. A file access hook part 107 manages the whole ciphering and deciphering processings of the file in the terminal 101, which are a processing for snatching a file access request from an application program 106 to the storage device 111 and the obtaining of a user key from a key managing server 112, etc. Besides, a system key for ciphering communication data exchanged between the terminal 101 and the key managing server 112 through the use of a network 122 and the key information record of a user 123 are preserved in the key database 121 of the key managing server 112.



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-149414

(43)公開日 平成11年(1999) 6月2日

(51)Int.Cl.⁸
G 0 6 F 12/14
G 0 9 C 1/00
H 0 4 L 9/08
9/32

識別記号
3 2 0
6 3 0

F I
G 0 6 F 12/14
G 0 9 C 1/00
H 0 4 L 9/00
3 2 0 B
6 3 0 B
6 0 1 B
6 0 1 A
6 7 3 C

審査請求 有 請求項の数17 O L (全 21 頁)

(21)出願番号 特願平10-199966
(22)出願日 平成10年(1998) 7月15日
(31)優先権主張番号 特願平9-247379
(32)優先日 平9(1997) 9月12日
(33)優先権主張国 日本 (J P)

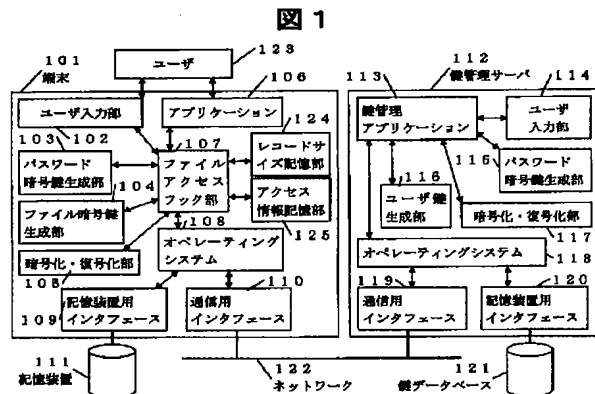
(71)出願人 000233055
日立ソフトウェアエンジニアリング株式会
社
神奈川県横浜市中区尾上町6丁目81番地
(72)発明者 宮崎 博
神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内
(72)発明者 鮫島 吉喜
神奈川県横浜市中区尾上町6丁目81番地
日立ソフトウェアエンジニアリング株式会
社内
(74)代理人 弁理士 秋田 収喜

(54)【発明の名称】 データの保存方法およびシステム並びにデータ保存処理用記録媒体

(57)【要約】

【課題】 ユーザが煩わしい操作を行なうことなく記憶装置内のデータの盗み見や改変を防止することができ、しかもコンピュータ自身が盗難されたとしても記憶装置内のデータを盗聴から防止可能にすること。

【解決手段】 アプリケーションプログラムからオペレーティングシステムへ発行する記憶装置へのデータ読み込み・書き出し要求を横取りし、書き出し要求の際にはデータを暗号化するための暗号鍵をネットワークで接続している鍵管理コンピュータから取り寄せ、この暗号鍵でデータを暗号化して記憶装置に保存し、また、読み込み要求の際には記憶装置から暗号化データを読み出し、この暗号化データを復号するための鍵を鍵管理コンピュータから取り寄せ、この暗号鍵で暗号化データを復号して取り出した元のデータをアプリケーションプログラムに渡す。



【特許請求の範囲】

【請求項1】 アプリケーションプログラムに従った処理を実行するコンピュータと、該コンピュータに接続された記憶装置とを備え、前記アプリケーションプログラムが処理したデータを前記記憶装置に書き込んで保存するデータの保存方法において、

前記アプリケーションプログラムからのデータの書き込み要求に対し、書き込み対象となるデータを前記記憶装置に書き込む前に、当該コンピュータとネットワークを介して接続された鍵管理コンピュータから暗号鍵を取り寄せ、該暗号鍵で書き込み対象のデータを暗号化して前記記憶装置に書き込み、

前記アプリケーションプログラムからの読み出し要求に対しては、前記記憶装置から暗号化されているデータを読み出した後に、前記ネットワークを介して前記鍵管理コンピュータから暗号鍵を取り寄せ、該暗号鍵で暗号化されたデータを復号し、その復号されたデータを読み出し要求を行なった前記アプリケーションプログラムに転送することを特徴とするデータの保存方法。

【請求項2】 前記データは、コンピュータが内部で生成した暗号鍵データを前記鍵管理コンピュータから取り寄せた暗号鍵で暗号化した暗号化暗号鍵データと、前記内部生成の暗号鍵データによりデータ本体を暗号化した暗号化データ本体とからなり、前記アプリケーションプログラムからのデータの書き込み要求に対し、書き込み対象となるデータを前記記憶装置に書き込む場合、コンピュータが内部で生成した暗号鍵データを前記鍵管理コンピュータから取り寄せた暗号鍵で暗号化した暗号化暗号鍵データを作成し、その暗号化暗号鍵データを、前記データ本体を前記内部生成の暗号鍵データで暗号化した暗号化データ本体に付加して前記記憶装置に書き込み、前記アプリケーションプログラムからの読み出し要求に対しては、前記記憶装置から暗号化データ本体に付加されている前記暗号化暗号鍵データを前記ネットワークを介して前記鍵管理コンピュータから取り寄せた暗号鍵で復号し、その結果得られた暗号鍵データを用いて前記暗号化データ本体を復号し、その復号されたデータ本体を読み出し要求を行なった前記アプリケーションプログラムに転送することを特徴とする請求項1記載のデータの保存方法。

【請求項3】 前記鍵管理コンピュータは、前記コンピュータを使用するユーザ毎に暗号鍵を管理するものであり、前記コンピュータは当該鍵管理コンピュータから前記暗号鍵を取り寄せる際には、データの読出しまたは書き込み要求を行ったアプリケーションプログラムを操作しているユーザに対して該ユーザを識別するための識別情報を入力させ、その識別情報の正しさを確認した後に、当該ユーザに対応した暗号鍵を前記鍵管理コンピュータから取り寄せることを特徴とする請求項1または2記載のデータ保存方法。

【請求項4】 前記識別情報をユーザが指定した時間の間、前記コンピュータの内部メモリに一時的に保存しておき、次のデータの読出しまたは書き込み要求があった際にはユーザによる識別情報を入力させることなく前記内部メモリに一時的に保存しておいた前記識別情報の正しさを確認した後に当該ユーザの暗号鍵を鍵管理コンピュータから取り寄せることを特徴とする請求項3記載のデータ保存方法。

【請求項5】 前記鍵管理コンピュータは、前記コンピュータを使用するユーザ毎に暗号鍵を管理するものであり、前記コンピュータは当該鍵管理コンピュータから前記暗号鍵を取り寄せる際には、データの読出しまたは書き込み要求を行ったアプリケーションプログラムを操作しているユーザに対して該ユーザを識別するための識別情報を入力させ、その識別情報を前記鍵管理コンピュータに送り、該鍵管理コンピュータで識別情報の正しさを確認させ、当該ユーザに対応した暗号鍵を取り寄せることを特徴とする請求項1または2記載のデータ保存方法。

【請求項6】 前記識別情報をユーザが指定した時間の間、コンピュータの内部メモリに一時的に保存しておき、次のデータの読出しまたは書き込み要求があった際には前記内部メモリに一時的に保存しておいた前記識別情報を鍵管理コンピュータに送り、鍵管理コンピュータで識別情報の正しさを確認させ、当該ユーザに対応した暗号鍵を取り寄せることを特徴とする請求項5記載のデータ保存方法。

【請求項7】 前記鍵管理コンピュータから取り寄せた暗号鍵をユーザが指定した時間の間、前記コンピュータの内部メモリに一時的に保存しておき、次のデータの読出しまたは書き込み要求があった際には、前記内部メモリに保存しておいた暗号鍵を使用して書き込み対象のデータの暗号化または読出し対象のデータの復号を行なうことを特徴とする請求項1乃至6記載のいずれかのデータ保存方法。

【請求項8】 前記記憶装置内において、ユーザが予め指定したファイル内のデータのみ、あるいは指定したディレクトリ内にある全てのファイルのデータのみに対して、暗号化または復号化を行うことを特徴とする請求項1乃至7記載のいずれかのデータ保存方法。

【請求項9】 アプリケーションプログラムに従った処理を実行するコンピュータと、該コンピュータに接続された記憶装置と、前記コンピュータにネットワーク接続される鍵管理コンピュータとから成り、前記アプリケーションプログラムが処理したデータを前記記憶装置に書き込んで保存するコンピュータシステムであって、前記コンピュータは、

前記アプリケーションプログラムからのデータの書き込み要求に対し、書き込み対象となるデータを前記記憶装置に書き込む前に、当該コンピュータとネットワークを

10

20

30

40

50

介して接続された前記鍵管理コンピュータから暗号鍵を取り寄せる第 1 の手段と、

この第 1 の手段で取り寄せた暗号鍵により書き込み対象のデータを暗号化して前記記憶装置に書き込む第 2 の手段と、

前記アプリケーションプログラムからの読み出し要求に対して、前記記憶装置から暗号化されているデータを読み出す第 3 の手段と、

前記ネットワークを介して前記鍵管理コンピュータから暗号鍵を取り寄せる第 4 の手段と、

この第 4 の手段で取り寄せた前記暗号鍵で前記第 3 の手段で読出した暗号化データを復号し、その復号されたデータを読み出し要求を行なった前記アプリケーションプログラムに転送する第 5 の手段と、を備えることを特徴とするコンピュータシステム。

【請求項 10】 アプリケーションプログラムに従った処理を実行するコンピュータと、該コンピュータに接続された記憶装置とを備えたコンピュータシステムにおいて前記アプリケーションプログラムが処理したデータを前記記憶装置に書き込んで保存するためのデータ保存処理用プログラムを記録した記録媒体であって、

前記アプリケーションプログラムからのデータの書き込み要求に対し、書き込み対象となるデータを前記記憶装置に書き込む前に、当該コンピュータとネットワークを介して接続された前記鍵管理コンピュータから暗号鍵を取り寄せる第 1 の処理と、

この第 1 の処理で取り寄せた暗号鍵により書き込み対象のデータを暗号化して前記記憶装置に書き込む第 2 の処理と、

前記アプリケーションプログラムからの読み出し要求に対して、前記記憶装置から暗号化されているデータを読み出す第 3 の処理と、

前記ネットワークを介して前記鍵管理コンピュータから暗号鍵を取り寄せる第 4 の処理と、

この第 4 の処理で取り寄せた前記暗号鍵で前記第 3 の処理で読出した暗号化データを復号し、その復号されたデータを読み出し要求を行なった前記アプリケーションプログラムに転送する第 5 の処理と、から成るデータ保存処理用プログラムが記録されていることを特徴とするデータ保存処理用記録媒体。

【請求項 11】 暗号化する際にファイルをあらかじめ決めた一定長のレコードに区切り、各レコード内で暗号連鎖処理を区切ることを特徴とする請求項 1 から 8 記載のいずれかのデータ保存方法。

【請求項 12】 前記レコード長をアプリケーションプログラムごとに決めることを特徴とする請求項 11 記載のデータ保存方法。

【請求項 13】 前記アプリケーションプログラムごとのレコード長として書き込み時のオーバーヘッドを最小化する値を求めるために、アプリケーションプログラムの

ファイルアクセス時の読み書き位置とそのアクセス長を記録しておいて、この記録からオーバーヘッドを最小化するレコード長を求めることを特徴とする請求項 11 または 12 記載のデータ保存方法。

【請求項 14】 前記レコード長をオペレーティングシステムが提供するファイルキャッシュ単位から決めることを特徴とする、請求項 11 乃至 13 記載のいずれかのデータ保存方法。

10 【請求項 15】 前記レコード長を記憶装置内のディスク上のセクタサイズから決めることを特徴とする請求項 11 乃至 14 記載のいずれかのデータ保存方法。

【請求項 16】 前記コンピュータシステムにおけるコンピュータは、さらに、

アプリケーションプログラムからの書き込み要求と読み込み要求に含まれるアクセス位置とアクセス長を該アプリケーションプログラムごとに記憶する第 6 の手段と、第 6 の手段で記憶したアプリケーションプログラムごとのアクセス位置とアクセス長の情報から、書き込み時のオーバーヘッド時間を最少にするレコード長を求める第 7 10 の手段と、

第 7 の手段で計算したレコード長をアプリケーションプログラムごとに記憶する第 8 の手段と、

ファイルの暗号連鎖処理を開始するために使う初期化データを生成してファイルに読み書きする第 9 の手段と、

第 2 の手段で書き込み要求データの暗号化を実行する際に、第 8 の手段で記憶しておいたレコード長を読み出し、

該書き込み要求データをレコード長に区切り、該書き込み要求データのレコードに対応するレコード鍵を第 1 の手段で得た暗号鍵から求め、該書き込み要求データのレコードに対応するレコード初期化データを第 9 の手段で得た初期化データから求め、暗号連鎖処理を行ないながら該書き込み要求データを暗号化する第 10 の手段と、

第 5 の手段で読み出したデータの復号化を実行する際に、第 8 の手段で記憶しておいたレコード長を読み出し、該読み出しデータをレコード長に区切り、該読み出しデータのレコードに対応するレコード鍵を第 4 の手段で得た暗号鍵から求め、該読み出しデータのレコードに対応するレコード初期化データを第 9 の手段で得た初期化データから求め、暗号連鎖処理を行ないながら該読み出しデータを復号化する第 11 の手段と、を備えることを特徴とする請求項 9 記載のコンピュータシステム。

【請求項 17】 前記データ保存処理用データプログラムは、さらに、アプリケーションプログラムからの書き込み要求と読み込み要求に含まれるアクセス位置とアクセス長を該アプリケーションプログラムごとに記憶する第 6 の処理と、第 6 の処理で記憶したアプリケーションプログラムごとのアクセス位置とアクセス長の情報から、書き込み時のオーバーヘッド時間を最少にするレコード長を求める第 7 40 の手段と、

アプリケーションプログラムからの書き込み要求と読み込み要求に含まれるアクセス位置とアクセス長を該アプリケーションプログラムごとに記憶する第 6 の処理と、

第 6 の処理で記憶したアプリケーションプログラムごとのアクセス位置とアクセス長の情報から、書き込み時のオーバーヘッド時間を最少にするレコード長を求める第 7 50 の手段と、

アプリケーションプログラムからの書き込み要求と読み込み要求に含まれるアクセス位置とアクセス長を該アプリケーションプログラムごとに記憶する第 6 の処理と、第 6 の処理で記憶したアプリケーションプログラムごとのアクセス位置とアクセス長の情報から、書き込み時のオーバーヘッド時間を最少にするレコード長を求める第 7

の処理と、
第 7 の処理で計算したレコード長をアプリケーションプログラムごとに記憶する第 8 の処理と、
ファイルの暗号連鎖処理を開始するために使う初期化データを生成してファイルに読み書きする第 9 の処理と、
第 2 の処理で書き込み要求データの暗号化を実行する際に、第 8 の処理で記憶しておいたレコード長を読み出し、該書き込み要求データをレコード長に区切り、該書き込み要求データのレコードに対応するレコード鍵を第 1 の処理で得た暗号鍵から求め、該書き込み要求データのレコードに対応するレコード初期化データを第 9 の処理で得た初期化データから求め、暗号連鎖処理を行ないながら該書き込み要求データを暗号化する第 10 の処理と、
第 5 の処理で読み出したデータの復号化を実行する際に、第 8 の処理で記憶しておいたレコード長を読み出し、該読み出しデータをレコード長に区切り、該読み出しデータのレコードに対応するレコード鍵を第 4 の処理で得た暗号鍵から求め、該読み出しデータのレコードに対応するレコード初期化データを第 9 の処理で得た初期化データから求め、暗号連鎖処理を行ないながら該読み出しデータを復号化する第 11 の処理と、から成るデータ保存処理用プログラムが記録されていることを特徴とする請求項 10 記載のデータ保存処理用記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、アプリケーションプログラムに従った処理を実行するコンピュータと、該コンピュータに接続された記憶装置とを備えたコンピュータシステムにおいて前記アプリケーションプログラムが処理したデータを前記記憶装置に書き込んで保存するためのデータ保存方法およびシステム並びにデータ保存処理用記録媒体に係り、特に、コンピュータ上で動作しているアプリケーションプログラムから記憶装置へのデータ読み出し・書き込み要求があった時に、そのコンピュータとネットワークで接続された鍵管理コンピュータからアプリケーションプログラムを操作しているユーザの暗号鍵を取り寄せ、その暗号鍵を用いて書き込みデータの暗号化および読出しデータの復号を行なうデータ保存方法およびシステム並びにデータ保存処理用記録媒体に関するものである。

【0002】

【従来の技術】従来、磁気ディスク装置等の記憶装置を備えたコンピュータシステムにあっては、コンピュータ内部で動作しているアプリケーションプログラムが処理したデータは、そのまま記憶装置に保存されていた。このため、記憶装置上に保存されているデータは、データ作成に使ったコンピュータや、さらにこの記憶装置がネットワークに接続している場合にはネットワーク上のコンピュータから、盗み見たり書き換えたりすることがで

き、機密漏洩やデータ改変などの恐れがあった。

【0003】そこで、記憶装置を管理しているオペレーティングシステムに、記憶装置上に保存しているデータに対して、どのユーザがデータを読み出したり書き換えたりして良いのかというアクセス制御機能を具備させ、アクセスの権利を持たないユーザからのアクセスを許可しない仕組みを講じたものがある。しかし、記憶装置に対してはオペレーティングシステムを介さずに直接アクセスすることもできるので、上記アクセス制御機能でもデータの盗み見や改変を完全に防ぐことはできなかった。そのため、記憶装置に保存するデータを暗号化したり、改変があったことを検知するための電子署名を付けたりして、これらの不正行為を防ぐ対策を取ってきた。

【0004】ところで、データを暗号化したり復号化したりする際には、データを 8 ビットや 64 ビットの基本単位に区切ってから暗号化・復号化する。しかし、このままでは同じ内容の基本単位は同じ暗号文に暗号化されてしまうので、暗号文解読の手がかりとなってしまう。そのため、データ内のある位置の基本単位を暗号化する際には、一つ前の位置のすでに暗号化されている基本単位を使って何らかの処理（例えば、排他的論理和）をしてから暗号化する。このような基本単位の連鎖処理を行うと、解読の手がかりを消すことができる。なお、データの先頭の基本単位を暗号化する際には、別に用意する初期化データを使って連鎖処理をする。上記の暗号連鎖処理については、例えば、「暗号とデータセキュリティ」（D. E. R. デニング著、上園忠弘他訳、培風館発行）に述べてある。

【0005】

【発明が解決しようとする課題】しかし、データを暗号化して保存する際には、データを記憶装置上に一旦保存してから、あるいは記憶装置に保存してあるデータをユーザが明示的に指定して暗号化を行っていた。同様に、暗号化データを復号する際にも、ユーザが明示的に復号の指定をして元のデータに復号していた。このため、ユーザがいちいちデータの暗号化・復号化の指示を行うことになり、操作が煩わしいという問題があった。特に、暗号化してあるデータを一度復号して修正してから再び暗号化するというような操作を行う場合には、再暗号化の操作を忘れる可能性があり、再暗号化の操作を忘れた場合には記憶装置内のデータの盗み見や改変に対抗できないという問題があった。したがって、データの暗号化・復号化が必要な時に、必ずしもユーザによる直接の指示がなくても、暗号化・復号化を行うことができる仕組みが必要となっている。

【0006】一方、現在は携帯用の小型端末としてのコンピュータが広く普及しており、どのような場所でも携帯用コンピュータを操作してデータ処理を行うといった状況が増えてきている。そのため、例えば会社内で機密情報を携帯用コンピュータに付属した記憶装置に保存し

ておき、社外でもその情報にアクセスする、というような状況も出てきている。しかし、このような携帯用コンピュータを置き忘れたり盗まれたりすると、機密漏洩の恐れがある。そのため、データを暗号化して保存しておけばすぐには元のデータを取り出すことはできない。しかし、データを暗号化するために使った暗号鍵は、通常、ユーザが入力するパスワードなどで暗号化し、暗号化データと同じ記憶装置に保存されているので、パスワードを類推されてしまえば、暗号鍵を取り出して暗号化データを復号できてしまう。すなわち、暗号鍵が何らかの形で暗号化されているとはいえ、暗号化データと暗号鍵を同じ記憶装置に保存していることが、機密データを保護する上で問題となっている。

【0007】本発明はこれらの問題点を解決するためになされたものであり、その第1の目的は、ユーザによる暗号化・復号化の直接の指示がなくても、アプリケーションプログラムからのデータの書き出し要求があったならば、そのデータを自動的に暗号化して記憶装置に保存し、データの読み込み要求があったら、記憶装置に保存してある暗号化データを自動的に復号してアプリケーションプログラムに渡すことにより、ユーザが煩わしい操作を行なうことなく記憶装置内のデータの盗み見や改変を防止することができ、しかもデータの暗号化・復号化に必要な暗号鍵を、暗号化データを保存する記憶装置が接続しているコンピュータとは異なる鍵管理コンピュータから取り寄せることにより、暗号化データをさらに強固に保護することができるデータ保存方法およびシステム並びに保存処理用記録媒体を提供することにある。

【0008】また、上記のようなアプリケーションプログラムからの書き出し要求、読み込み要求を受けて自動的に暗号化・復号化する場合、次のような問題が生じる。

【0009】アプリケーションプログラムはファイル内の任意の位置からデータの読み書きを行う。自動的な暗号化方法を行う際に暗号基本単位の連鎖処理を行っていると、ファイル内のある位置でデータの上書きが起きた時にその位置以降のすべての暗号基本単位を暗号化し直す処理が必要となり、上書き処理終了までの時間が大きくなってしまふ。

【0010】このような再暗号化処理が必要となる部分を少なくするため、ファイルを暗号基本単位より大きい一定長のレコードで区切り、このレコードごとに上記初期化データを用意する方式が考えられている。この方式では、ファイル内のある位置でデータの上書きが起きた場合、再暗号化処理を上書きのあったレコードの終端までしか行わないようしている。このようにすることで、再暗号化が必要となる部分を限定することができる。

【0011】しかし、ファイルを一定長のレコードに分割することにより、データ上書きの際の再暗号化処理を少なくすることはできるが、どの程度のレコード長が良いかは簡単に決めることはできない。

【0012】レコード長が長いと、アプリケーションプログラムがデータを上書きする時に必要となる再暗号化処理の長さが大きくなってしまふ。

【0013】レコード長が短いと、アプリケーションプログラムがデータを読み書きする際に複数のレコードにまたがってアクセスすることが多くなり、各レコードの初期化データを準備する回数が増えてしまい、その処理時間がかえってオーバーヘッドになってしまふ。

【0014】また、アプリケーションプログラムごとに読み書きする単位長が異なっているため、あるアプリケーションプログラムにとって最適なレコード長が他のアプリケーションプログラムにとっても最適とは限らない。

【0015】本発明は上記問題点も解決するためになされたものであり、その第2の目的は、アプリケーションプログラムそれぞれについて、その読み書きする長さや読み書きを開始する位置を記録しておき、この記録を用いて、上書きの際の再暗号化処理や初期化データの準備からなるオーバーヘッド時間を最小化するレコードサイズを求めることにより、データ上書きの際の処理時間を短くすることができるデータ保存方法およびシステム並びに保存処理用記録媒体を提供することにある。

【0016】

【課題を解決するための手段】上記目的を達成するため、本発明のデータ保存方法は、アプリケーションプログラムからオペレーティングシステムへ発行する記憶装置へのデータ読み込み・書き出し要求を横取りし、書き出し要求の際にはデータを暗号化するための暗号鍵をネットワークで接続している鍵管理コンピュータから取り寄せ、この暗号鍵でデータを暗号化して記憶装置に保存し、また、読み込み要求の際には記憶装置から暗号化データを読み出し、この暗号化データを復号するための鍵を鍵管理コンピュータから取り寄せ、この暗号鍵で暗号化データを復号して取り出した元のデータをアプリケーションプログラムに渡すようにしたことを特徴とするものである。また、鍵管理コンピュータに対して暗号鍵を送ってもらう要求を出す前に、アプリケーションプログラムを操作しているユーザに対してそのユーザを識別するための情報を入力させ、この情報の正しさをユーザが操作しているコンピュータ上で確認した後にユーザの鍵を鍵管理コンピュータから取り寄せる、あるいはユーザが入力した識別情報を鍵管理コンピュータに送り、鍵管理コンピュータでは受け取った識別情報の正しさを確認した後にそのユーザの暗号鍵を送り返すことで、ユーザごとの鍵管理を行うことを特徴とするものである。また、予めユーザがファイル保存の際には自動的に暗号化して欲しいファイルやディレクトリを指定することで、指定したファイルあるいは指定したディレクトリ以下にあるファイルだけに対して、暗号化や復号化を行うことを特徴とするものである。また、オペレーティングシス

テムがアプリケーションプログラムからのデータ読み込み・書き込み要求を受け取ると、そのデータのファイル内での位置とデータ長を記録しておき、十分な数の記録を取った後、アプリケーションプログラムごとに再暗号化処理と初期化データ処理のオーバーヘッドを最小化するレコード長を計算し、以降の暗号化・復号化の際には計算で求めたレコード長を用いることを特徴とするものである。

【0017】このような手段により、暗号化や復号化の指示をユーザが明示的に行うことなしに記憶装置に保存するデータを自動的に暗号化し、読み出す時には自動的に復号することができ、わずらわしい操作をユーザに強いことがない。また、記憶装置に保存するデータを自動的に保存するので、ユーザがデータの暗号化を忘れるという問題も起きない。また、暗号化して保存する場所としてフロッピーディスクのような記憶媒体を表わすディレクトリも指定できるので、重要な情報をコピーされて持ち出されることを防ぐこともできる。また、暗号化したデータと暗号鍵が物理的に離れた記憶装置に保存されることにより、暗号化した機密データを保存してある携帯端末が盗まれたとしても、解読される危険性はこれらのデータが同一の記憶装置に保存されている場合に比べて小さくなる。例えば、携帯端末に暗号化データを保存しておき、社内ですそのデータにアクセスしたい時は社内LANに接続し、社外からは社内ネットワークヘリモートアクセスして利用するようにしておけば、この携帯端末を不正に入手したユーザは、ユーザの識別情報を類推するだけでなく、リモートアクセス時のユーザ認証作業も行う必要があるので安全性がより高くなる。また、アプリケーションプログラムごとに、1回の上書き要求における再暗号化の長さ、および初期化データの生成処理からなるオーバーヘッドを最小にするレコード長を求めることができるので、データ上書きの際の処理時間をそれぞれのアプリケーションプログラムごとに高速化できる。

【0018】また、アプリケーションからオペレーティングシステムへのデータ読み出し・書き込み要求を横取りするための仕組みは、オペレーティングシステムが提供している場合には、既存のアプリケーションプログラムやオペレーティングシステムを書き換える必要がなく、しかも新たにハードウェアを置き換える必要もなく、最小限のコストでデータのセキュリティを高めることが可能である。

【0019】

【発明の実施の形態】以下、本発明の実施形態に基づき詳細に説明する。図1は、本発明の実施形態を示すシステム構成図である。本実施形態は、ユーザが端末上で動作しているアプリケーションプログラムで作成したデータを、鍵管理サーバ（鍵管理コンピュータ）から取り寄せる自分の鍵で暗号化して端末に接続の記憶装置に保存

する場合と、記憶装置から読み出した暗号化データを鍵管理サーバから取り寄せる自分の鍵で復号化して平文のデータをアプリケーションに渡す場合のものである。

【0020】図1に示すシステムは、大別すると、ユーザ123が操作する端末101と、この端末101に接続された記憶装置111と、データの暗号化のための鍵を管理する鍵管理サーバ（鍵管理コンピュータ）112と、暗号化のための鍵を格納した鍵データベース121と、端末101と鍵管理サーバ112とを接続するネットワーク122とから構成されている。

【0021】端末101は、据置き型あるいは携帯用のコンピュータで構成されるものであり、その内部には、ユーザ入力部102、パスワード暗号鍵生成部103、ファイル暗号鍵生成部104、暗号鍵・復号化部105、アプリケーションプログラム106、ファイルアクセスフック部107、オペレーティングシステム108、記憶装置用インタフェース109、通信用インタフェース110、レコードサイズ記憶部124、アクセス情報記憶部125を備え、ユーザ123はアプリケーションプログラム106で作成したデータファイルを記憶装置111に保存したり、読み出したりすることができる。

【0022】ユーザ入力部102は、ファイルアクセスフック部107からの指示によって、ユーザ123にユーザ名とパスワードを入力させるためのものである。パスワード暗号鍵生成部103は、ユーザ123が入力したパスワードから暗号鍵を生成するためのものである。パスワード暗号鍵は、ユーザ123の認証情報やシステム全体で使う暗号鍵を暗号化して記憶装置111に保存するものである。ファイル暗号鍵生成部104は、記憶装置111に保存するファイルの暗号化に使う暗号鍵をランダムに生成するものである。暗号化・復号化部105は、ファイルを実際に暗号化・復号化するものである。ユーザ123がデータファイルを作成するためのアプリケーションプログラム106は、ワープロや表計算などのソフトである。

【0023】ファイルアクセスフック部107は、アプリケーションプログラム106から記憶装置111へのファイルアクセス要求を横取りする処理と、鍵管理サーバ112からユーザ鍵を取り寄せるなど、端末101でのファイルの暗号・復号処理全般を管理するものである。端末101上で動作しているオペレーティングシステム108は、記憶装置用インタフェース109、通信用インタフェース110、レコードサイズ記憶部124、アクセス情報記憶部125を管理して、記憶装置111へのアクセスや、ネットワーク122を介した通信などの動作環境をアプリケーションプログラム106やファイルアクセスフック部107に提供している。

【0024】レコードサイズ記憶部124は、アプリケーションプログラムごとの暗号化・復号化する際のレコ

ードサイズを記憶するために使う。

【0025】アクセス情報記憶部125は、アプリケーションプログラムごとに、ファイル内で読み出しや書き込みを開始した位置と読み出し・書き込みデータの長さを複数保存しておくためのもので、暗号化・復号化する際のレコードサイズを計算するために使う。

【0026】なお、記憶装置111には、ユーザ123が作成したファイルだけでなく、ユーザ123がパスワードを入力した際の確認に使う認証トークンやシステムがデータの暗号化に使うシステム鍵も暗号化して保存してある。

【0027】一方、鍵管理サーバ112は、鍵管理アプリケーションプログラム113、ユーザ入力部114、パスワード暗号鍵生成部115、ユーザ鍵生成部116、暗号化・復号化部117、オペレーティングシステム118、通信用インタフェース119、記憶装置用インタフェース120を備えており、端末101のユーザ123が本システムを使うための情報を生成したり、ユーザ鍵を管理する機能を担っている。

【0028】鍵管理アプリケーションプログラム113は、ユーザ123に配布する情報の生成や、端末101からのユーザ鍵要求メッセージの処理など、鍵管理サーバ112での暗号処理全般を管理している。ユーザ入力部114は、最初にユーザ123がユーザ鍵を生成する際に、鍵管理アプリケーションプログラム113からの指示によって、ユーザ123にユーザ名とパスワードを入力させるためのものである。パスワード暗号鍵生成部115は、ユーザ123が入力したパスワードから暗号鍵を生成するためのものである。ユーザ鍵生成部116は、ユーザ123用の暗号鍵をランダムに生成するためのものである。暗号化・復号化部117は、ユーザ123に配布する情報を暗号化したり、ユーザ123がユーザ鍵を要求してきた時にやり取りする電子メッセージを暗号化あるいは復号化したりするためのものである。

【0029】鍵管理サーバ112上で動作するオペレーティングシステム118は、通信用インタフェース119、記憶装置用インタフェース120を管理し、ネットワーク122を使った通信や鍵データベース121へのアクセスなどの環境を、鍵管理アプリケーションプログラム113に提供している。

【0030】鍵データベース121には、端末101と鍵管理サーバ112とがネットワーク122を使ってやりとりする通信データを暗号化するためのシステム鍵と、ユーザ123の鍵情報レコードが保存してあり、この鍵情報レコードは、図2に示すように、ユーザ名201とユーザ鍵202とから構成されている。

【0031】(ユーザ配布情報の作成) 図3は、本システムを使用するためにユーザ123へ配るデータを鍵管理サーバ112上で作成する手順を示すフローチャートである。まず、ユーザ入力部114からユーザ名とパス

ワードを入力させる(ステップ301)。次に、パスワード暗号鍵生成部115で、ユーザ123のパスワードに処理を加えてパスワード暗号鍵を生成する。処理方法としては、例えば、MD5アルゴリズムのような一方向関数を用いる(ステップ302)。続いて、ユーザ鍵生成部116において、ユーザ鍵をランダムに生成する(ステップ303)。ここで得られたユーザ鍵とユーザ名から、図2に示すような鍵情報レコードを作成し、鍵データベース121に保存する(ステップ304)。なお、図2はユーザ名が「A」であるユーザAの鍵情報レコードの例である。

【0032】次に、ユーザ123が入力したパスワードの正しさを検証するための認証トークンを作成する(ステップ305)。これは、ユーザが入力したパスワードをそのままデータとして保存しておくことはセキュリティ上危険であるため、暗号処理を利用して保存している。例えばUNIXでは、「オール0」のデータをユーザ123のパスワードを鍵として暗号化して保存しておく、後でユーザ123がパスワードを入力した際に同じ処理をして認証作業を行っている。本実施形態では、すべて「0」の6バイトデータをパスワード暗号鍵で暗号化して図4のような認証トークンを生成する。図4において、401は、この認証トークンの所有者を示すユーザ名(ここではユーザA)で、402は暗号処理を行った後のデータ値である。

【0033】続いて、本システムでネットワーク122上を流れるデータを暗号化するために使うシステム鍵を、やはりパスワード暗号鍵で暗号化する(ステップ306)。パスワード暗号鍵を使うことで、ユーザ123のパスワードを知っている者だけが、正しいシステム鍵を手に入れることができる。なお、システム鍵は、鍵管理サーバ112においてシステムの運用開始時に作成しておく。最後に、認証トークンと暗号化したシステム鍵とをフロッピーディスクなどにコピーしてユーザ123に渡す(ステップ307)。ユーザ123は受け取った情報を、自分が操作する端末101に接続してある記憶装置111にコピーして使用することになる。

【0034】(ファイルの暗号化) 図5は、ユーザ123(ここではユーザA)が端末101上で作成したデータを暗号化して記憶装置111に保存する際の手順を示すフローチャートである。本手順は、ユーザAがアプリケーションプログラム106で作成したデータを記憶装置111に保存しようとした時に開始する。まず、アプリケーションプログラム106がオペレーティングシステム108へのファイル保存要求を発行すると、オペレーティングシステム108がその要求を処理する前に、そのファイル保存要求をファイルアクセスフック部107が受け取る(ステップ501)。

【0035】続いて保存要求があったファイルを暗号化して保存するために、以下のようにして管理サーバ11

2からユーザ鍵を取り寄せる。まず、アプリケーションプログラム106を操作していたユーザAのユーザ名とパスワードをユーザ入力部102からユーザAに入力させるステップ502)。次に、入力されたパスワードからパスワード暗号鍵生成部103でパスワード暗号鍵を生成する(ステップ503)。

【0036】端末101は、入力されたパスワードが正しいかどうか確かめるため、管理サーバ112での認証トークンの生成と同じ処理を行い、記憶装置111から読み出した認証トークンと一致するかどうか検証する

(ステップ504)。一致しなければ、ユーザ名とパスワードの入力を繰り返させる。一致した場合は、次の手順に進む。

【0037】次に、記憶装置111から暗号化してあるシステム鍵を読み出し、パスワード暗号鍵を使ってシステム鍵を復号する(ステップ505、506)。なお、記憶装置111から読み出した認証トークンと暗号化システム鍵は、前もって鍵管理サーバ112において生成しておいたものである。

【0038】次に、鍵管理サーバ112とメッセージの交換を行い、ユーザ鍵を取り寄せる。なお、端末101と鍵管理サーバ112の間で交換するメッセージのフォーマットは、図6に示すようなものである。図6において、601は、メッセージの区別をするためのヘッダ部であり、「0」であれば端末101から鍵管理サーバ112へユーザ鍵を要求するユーザ鍵要求メッセージであることを示し、「1」であれば鍵管理サーバ112からの返答であることを示すユーザ鍵返答メッセージである。602は、メッセージ中に含まれるデータ本体であり、メッセージの種類により内容は異なる。603は、システム鍵を使ったヘッダ部601とデータ部602の電子署名である。

【0039】鍵管理サーバ112からユーザ鍵を取り寄せるため、端末101ではユーザ鍵要求メッセージを作成する(ステップ507)。端末101から鍵管理サーバ112へ送る情報は、ユーザ名とチャレンジデータである。チャレンジデータとは、ユーザ鍵要求メッセージを作成するたびにランダムに選ぶ整数で、鍵管理サーバ112が事前に決めた処理(例えば1を加える)をしたレスポンスデータをユーザ鍵返答メッセージに含める決まりにしておくことで、ユーザは受け取った返答メッセージが、直前に出した要求メッセージに対する返答であることを確認できる。まず、図7に示すようなユーザ名701とチャレンジデータ702からなるデータを構成し、システム鍵で暗号化する。次に図8のように、ヘッダ部801にメッセージがユーザ鍵要求メッセージであることを示す値「0」と、データ部802にユーザ名とチャレンジデータを暗号化した情報からなるデータの電子署名803とから構成されたユーザ鍵要求メッセージを生成する。そして生成したユーザ鍵要求メッセージを

鍵管理サーバ112に送り、返答メッセージが送られるのを待つ。この後、鍵管理サーバ112はユーザ鍵要求メッセージを処理し、ユーザ鍵返答メッセージを作成して端末101に送り返してくるが(ステップ508)、この手順の詳細は図13を用いて後述する。

【0040】端末101では、送り返されてきたユーザ鍵返答メッセージを受け取ると(ステップ509)、まず、転送途中での改変がなかったか、返答メッセージに含まれている電子署名を検査する(ステップ510)。ここでメッセージの改変があったことが分かった、ファイルの保存命令を出したアプリケーションプログラム106に保存の失敗を知らせ終了する(ステップ521)。改変がない場合には次の手順に進む。ユーザ鍵返答メッセージのデータ部はシステム鍵で暗号化されているので、システム鍵で復号する(ステップ511)。

【0041】ところで、ユーザ鍵返答メッセージには、ユーザ鍵を含むものと、鍵管理サーバ112で何らかのエラーが発生したためにユーザ鍵を含まないものの2種類があり、返答メッセージのデータ部の内容が異なっている。このデータ部の暗号化される前の内容は図9あるいは図10のようになっている。図9は、ユーザ鍵が返されてきた場合の内容であり、ユーザ名901、レスポンスデータ902、ユーザ鍵が含まれていることを示す値「0」903、ユーザ鍵904を含んでいる。図10は、何らかのエラーによりユーザ鍵が返されなかった場合の内容で、ユーザ名1001、レスポンスデータ1002、何らかのエラーが発生したことを示す値「1」1003を含んでいる。ここで1003には、エラーの具体的な種類を示すために「0」以外のどんな値を含めてもよい。なお、ユーザ鍵返答メッセージに含めるレスポンスデータは、ユーザ鍵要求メッセージに含まれていたチャレンジデータの値に「1」を加えたものとする。

【0042】端末101は、ユーザ鍵返答メッセージのデータ部を復号した後、ユーザ鍵の有無を検査する(ステップ512)。ユーザ鍵を含んでいない場合は、ファイルを暗号化して保存することができないので、保存の失敗をアプリケーションプログラム106に知らせて(ステップ521)終了する。ユーザ鍵が含まれていた場合には、実際にファイルを暗号化する手順に進む。

【0043】ここで新規ファイルへの書き込みであれば、ファイル暗号鍵とファイル初期化データをランダムに生成する(ステップ519)。そしてファイル暗号鍵をユーザ鍵で暗号化した後、暗号化したファイル暗号鍵とファイル初期化データを新規ファイルのヘッダとして保存して(ステップ520)、次の手順に進む。ファイル暗号鍵は、ファイルを分割したレコードごとに用意する暗号鍵を生成するために使う。ファイル初期化データは、レコードごとの初期化データを生成するために使う。

【0044】もし既存ファイルへの書き込みであれば、

書き込み対象のファイルのヘッダから暗号化されたファイル暗号鍵とファイル初期化データを読み込む（ステップ 5 1 4）。そして、暗号化されたファイル暗号鍵をユーザ鍵で復号化してファイル暗号鍵を得る（ステップ 5 1 5）。次に、書き込み要求のあったデータを暗号化する（ステップ 5 1 6）。なお、暗号化の手順の詳細は図 1 4 で説明する。

【0 0 4 5】上記のようにして得た暗号化されたデータは、アプリケーションプログラム 1 0 6 の書き込み要求に対応した暗号化ファイル内の位置に書き込む。そして、暗号化されたファイル暗号鍵を 1 1 0 1、ファイル初期化データを 1 1 0 2、暗号化されたファイル本体を 1 1 0 3 とし、ユーザ鍵を使ってこれらの署名を生成したものを 1 1 0 4 として図 1 1 に示すような暗号化ファイル 1 1 0 0 を作成し、記憶装置 1 1 1 に保存する。従って、記憶装置 1 1 1 には、アプリケーションプログラム 1 0 6 が生成したファイルを暗号化した暗号化ファイル本体 1 1 0 3 に対し、ファイル暗号鍵を暗号化したデータ 1 1 0 1、ファイル初期化データ 1 1 0 2 と署名 1 1 0 4 が付加されて記憶されることになる。

【0 0 4 6】最後に、保存の終了をファイルの保存要求を出したアプリケーションプログラム 1 0 6 に知らせ（ステップ 5 1 8）、本手順を終了する。

【0 0 4 7】（ファイルの復号化）図 1 2 は、ユーザ A が記憶装置 1 1 1 に保存してある暗号化データを端末 1 0 1 上で動作するアプリケーションプログラム 1 0 6 で読み出す際の手順を示すフローチャートである。本手順は、ユーザ A が以前に暗号化して記憶装置 1 1 1 に保存したデータをアプリケーションプログラム 1 0 6 に読み込もうとした時に開始する。アプリケーションプログラム 1 0 6 がオペレーティングシステム 1 0 8 へのファイル読み出し要求を発行すると、オペレーティングシステム 1 0 8 がその要求を処理する前に、そのファイル読み出し要求ファイルアクセスフック部 1 0 7 が受け取る（ステップ 1 2 0 1）。ファイルアクセスフック部 1 0 7 は、読み出し要求で指定されたファイルを読み出すようにオペレーティングシステム 1 0 8 に指示し、記憶装置 1 1 1 から記憶装置用インタフェース 1 0 9 を介して、暗号化されているファイルを読み出す（ステップ 1 2 0 2）。

【0 0 4 8】次に、読み込んだファイルを復号するために、ファイルを暗号化すると同様の手順でユーザの認証を行い、鍵管理サーバ 1 1 2 からユーザ鍵を取り寄せる。このユーザ鍵を取り寄せるまでの手順に相当するステップ 1 2 0 3 ~ 1 2 1 3 は、図 5 のステップ 5 0 2 ~ 5 1 2 と全く同様である。この際に、鍵管理サーバ 1 1 2 からのユーザ鍵返答メッセージの改変を検知したり（ステップ 1 2 1 1）、あるいはユーザ鍵を得ることができなかった場合には（ステップ 1 2 1 3）、読み込んだファイルを復号することができないので、読み出しの

失敗をアプリケーションプログラム 1 0 6 に知らせて（ステップ 1 2 1 9）本手順を終了する（ステップ 1 2 1 8）。ユーザ鍵を得ることができた場合のみ、以下のようにして暗号化したファイルを復号する。ここで読み込み対象のファイルは図 1 1 に示すようなフォーマットであり、1 1 0 1 はファイル暗号鍵をユーザ鍵で暗号化したもの、1 1 0 2 はファイル初期化データ、1 1 0 3 は暗号化されたファイル本体、1 1 0 4 はこれら 3 つのデータ 1 1 0 1 ~ 1 1 0 3 のユーザ鍵による電子署名である。

【0 0 4 9】図 1 2 において、ユーザ鍵が含まれていた場合、暗号化ファイルが改変されていないか、署名 1 1 0 4 の確認を行なう（ステップ 1 2 1 4）。ここで改変されていることがわかったら、ファイル読み出しの失敗をアプリケーションプログラム 1 0 6 に通知して（ステップ 1 2 1 9）本手順を終了する。

【0 0 5 0】改変されていないければ、まず暗号化ファイルから暗号化されたファイル暗号鍵 1 1 0 1 とファイル初期化データ 1 1 0 2 を読み込む（ステップ 1 2 1 5）。

【0 0 5 1】次に、鍵管理サーバ 1 1 2 から得たユーザ鍵を使い、暗号化ファイル暗号鍵 1 1 0 1 を復号し、元のファイル暗号鍵を得る（ステップ 1 2 1 6）。

【0 0 5 2】続いて、暗号化されているファイル本体 1 1 0 3 からアプリケーションプログラム 1 0 6 の要求に対応した暗号化データを読み込み、復号して得たファイル暗号鍵を使って復号する（ステップ 1 2 1 1）。なお、復号の手順の詳細は図 1 9 で説明する。

【0 0 5 3】最後に、復号して得たデータをアプリケーションプログラム 1 0 6 に渡して（ステップ 1 2 1 8）本手順を終了する。

【0 0 5 4】（鍵管理アプリケーションプログラムの処理）図 1 3 は、鍵管理サーバ 1 1 2 上で動作する鍵管理アプリケーションプログラム 1 1 3 が、端末 1 0 1 からのユーザ A のユーザ鍵要求を処理する際の手順を示すフローチャートである。本手順は、ファイルアクセスフック部 1 0 7 からのユーザ鍵要求メッセージを受け取って開始する。鍵管理アプリケーションプログラム 1 1 3 が受け取るユーザ鍵要求メッセージは、図 8 に示したようなフォーマット構成になっている。鍵管理アプリケーションプログラム 1 1 3 は、端末 1 0 1 からユーザ鍵要求メッセージを受け取ると（ステップ 1 3 0 1）、最初に、メッセージに改変がなかったか調べるため、鍵データベース 1 2 1 にあらかじめ保存してあるシステム鍵を読み込み（ステップ 1 3 0 2）、要求メッセージ中の電子署名を確認する（ステップ 1 3 0 3）。もし改変があれば、以下に述べるユーザ鍵を鍵データベース 1 2 1 から取り出す作業を行わない。

【0 0 5 5】ユーザ鍵を鍵データベース 1 2 1 から取り出すには、まず暗号化・復号化部 1 1 7 において、ユー

10

20

30

40

50

ザ鍵要求メッセージをシステム鍵で復号し、図7に示したようなユーザ名701とチャレンジデータ702を取り出す(ステップ1304)。次に、ここで得たユーザ名を含む鍵情報レコードを鍵データベース121から取り出す。図2の示すような鍵情報レコードが見つければ、ユーザ鍵202を取り出す(ステップ1305)。

【0056】以上のようにして、ユーザ鍵要求メッセージの改変検査とユーザ鍵の取り出しを終えた後、端末101に返答するユーザ鍵返答メッセージを作成する(ステップ1306)。ここで、返答メッセージに含めるレスポンスデータは、要求メッセージに含まれていたチャレンジデータに「1」を加えた値であるが、ユーザ鍵を得ることができたかどうかで返答メッセージのデータ部に含める内容が変わる。

【0057】ユーザ鍵を得た場合には、図9に示したようにユーザ名901、レスポンスデータ902、ユーザ鍵を含んでいることを示す値「0」903、ユーザ鍵904から成るデータを暗号化・復号化部117上でシステム鍵を使い暗号化したものを、ユーザ鍵返答メッセージのデータ部に含める。ユーザ鍵を得ることができなかった場合には、図10に示したようにユーザ名1001、レスポンスデータ1002、ユーザ鍵を含んでいないことを示す値「1」から成るデータを暗号化・復号化部117上でシステム鍵を使い暗号化したものを、ユーザ鍵返答メッセージのデータ部に含める。ユーザ鍵返答メッセージの署名部には、ユーザ鍵返答メッセージであることを示す値「1」を持つヘッダ部と、上記のように作成したデータ部からなるデータを、システム鍵を使って作成した署名を含める。

【0058】以上のようにして作成したユーザ鍵返答メッセージを端末101に送り(ステップ1307)、本手順を終了する。

【0059】(暗号化詳細手順)図14は、アプリケーションプログラム106から書き込み要求のあったデータを暗号化する際の詳細手順を示すフローチャートである。本手順は、ファイル暗号鍵、ファイル初期化データ、書き込み要求データを用意できた時点で開始する。

【0060】図15は、アプリケーションからの書き込み要求である。1501は書き込み開始位置、1502は書き込みデータの長さ、1503は書き込みデータである。図15の書き込み要求では、ファイル内の位置1800から2799までの間に1503の書き込みデータを書き込むことを要求している。

【0061】図14において、まず、書き込み要求を行なったアプリケーションプログラムに対応するレコード長をレコードサイズ記憶部から読み込む(ステップ1401)。

【0062】レコードサイズ記憶部124は、図16に示すように、表計算プログラムやワープロプログラムなどの各種のアプリケーションプログラム1601ごとの

レコードサイズ10602を保存してある。もし、対応するレコード長が記録されていなかったら、レコードサイズ記憶部124の最後に記録されているデフォルトの長さを読み込む。ここでは、書き込み要求を行なったアプリケーションプログラムに対応する長さとして「512」を読み込むものとする。

【0063】もし、レコードサイズ記憶部124にレコード長が記録されていなかったら、ここでアプリケーションプログラムからの書き込み要求に含まれる、ファイル内の書き込み要求位置1501と書き込みデータの長さ1502を、このアプリケーションプログラム用のアクセス情報記憶部125に保存する(ステップ1403)。

【0064】アクセス情報記憶部125は、図17に示すように、書き込み、読み込み要求があるたびにその開始位置1701と書き込みデータの長さ1702をアプリケーションプログラムごとに記録したもので、複数記憶しておくことができる。図17では、図15の書き込み要求をアクセス情報記憶部125に保存した場合の様子を示している。ここで記録したデータは、一定時間後に暗号する際のレコード長を求めるために使う。

【0065】次に、書き込み要求の開始位置をレコード長で割り算し、その商をファイルの何番目のレコードへの書き込みであるかを示すレコードインデックス値とする(ステップ1404)。もし、複数のレコードにまたがる書き込みであれば、すべてのレコードについて計算しておく。図15に示す書き込み要求からレコードインデックスを求めると、書き込み開始位置「1800」をレコード長「512」で割った商は3、書き込み終了位置2799を512で割った商は5であるので、求めるレコードインデックス値は「3」、「4」、「5」である。

【0066】続いて、書き込み先のレコード用のレコード暗号鍵を、そのレコードのインデックス値とファイル暗号鍵を使って計算する(ステップ1405)。もし、レコードの先頭から書き込み要求を行なう必要があれば、そのレコード用の初期化データをファイル初期化データとレコードインデックス値から求める。これらの計算方法としては、例えば2つのデータを連結してハッシュ関数による演算を行なう方法が考えられる。図15の書き込み要求の場合、レコード3の途中からレコード5の途中までの書き込み要求であるので、「レコード3」のレコード暗号鍵の計算、および「レコード4」、「5」のレコード暗号鍵と初期化データの計算を行う。

【0067】次に、書き込むデータがすでにファイル内に保存されているデータの上書きであるかどうか検査する(ステップ1406)。ここで上書きでなければ、書き込みデータを暗号化用バッファにコピーし暗号化ステップに進む(ステップ1411)。

【0068】もし上書きであれば、書き込み開始位置で

暗号連鎖処理に必要な部分と再暗号化が必要な部分をファイルから暗号化用バッファに読み込む(ステップ1407)。暗号連鎖処理に必要な部分とは、書き込み開始位置でデータを暗号化する際に連鎖処理で使う暗号文のことで、書き込み開始位置の直前の暗号文であり、その長さは暗号連鎖処理の方法によって異なる。書き込み開始位置がレコードの先頭であれば、暗号連鎖処理に必要な部分としてレコード初期化データを用いるので、書き込み開始位置の直前部分をファイルから読み込む必要はない。本実施例における暗号連鎖処理に必要な長さ

10 を「8」とすると、図15の書き込み要求における暗号連鎖処理に必要な部分とは、ファイル内の位置「1792」から「1799」までの暗号文である。また、再暗号化が必要な部分とは、データの上書きによって暗号連鎖処理をやり直す必要のある部分のことで、書き込み終了位置の直後からこの書き込み終了位置を含むレコードの最後までである。図15の書き込み要求における再暗号化が必要な部分は、ファイル内の書き込み終了位置の直後「2800」から、この書き込み終了位置を含む「レコード5」の最後の位置「3071」までである。

20 図18に、ファイル内における、暗号連鎖処理に必要な部分と再暗号化が必要となる部分を示す。

【0069】続いて、暗号化用バッファ上にある再暗号化が必要な部分を、対応するレコード暗号鍵を使って復号しておく(ステップ1408)。図15の書き込み要求では、「レコード5」のレコード暗号鍵を使って復号する。そして、書き込み要求データを再暗号化のために復号したデータとつなげて暗号化用バッファ上に書き込む(ステップ1409)。

【0070】最後に、暗号化用バッファ上に暗号化するデータが用意できたら、そのデータに対応するレコード用の暗号鍵とファイルから読み込んだ暗号連鎖処理に必要なデータ、初期化データを使って暗号化して(ステップ1410)本手順を終了する。

【0071】(復号化詳細手順)図19は、アプリケーションプログラムから読み込み要求のあったデータを読み込んで復号する際の詳細手順を示すフローチャートである。本手順は、ファイル暗号鍵、ファイル初期化データを用意できた時点で開始する。

【0072】図20は、アプリケーションからの読み込み要求である。2001は読み込み開始位置、2002は読み込みデータの長さである。

【0073】図19において、まず、書き込み要求の時と同様に、読み込み要求を行なったアプリケーションプログラムに対応するレコード長をレコードサイズ記憶部から読み込む(ステップ1901)。もし、対応するレコード長がなければデフォルトの値を読み込み(ステップ1902)、読み込み要求位置2001とその長さ2002をこのアプリケーションプログラム用のアクセス情報記憶部125に記録する(ステップ1903)。レ

コードインデックスの計算とレコード用の暗号鍵、初期化データの計算も、書き込み要求の場合と同様に行なう(ステップ1904、1905)。

【0074】続いて、読み込み要求に対応するレコードをファイルから読み込み、レコード用暗号鍵、初期化データを使って復号して(ステップ1906)、本手順を終了する。

【0075】(レコード長の計算)図21は、レコード長を計算する際の詳細手順を示すフローチャートである。本手順は、あるアプリケーションプログラムにおいて、アクセス情報記憶部125に記録したアクセス情報が一定件数(例えば100件)を越えた時に開始する。ここで、レコード長としては2のべき乗を候補とする。アプリケーションプログラムやオペレーティングシステムのファイルアクセス単位、記憶装置上の物理的な記録単位は、処理の単純化のために2のべき乗の数を使う例が多いためである。なお、この手順を始める前に、暗号化速度、復号化速度、レコード用の暗号鍵の生成にかかる時間、初期化データの生成にかかる時間を求めておく。

【0076】図21において、まず、アクセス情報記憶部125に記録したアクセス長の平均値を求める(ステップ2102)。ここで、平均アクセス長として「140」バイトが得られたものとする。次に、平均アクセス長を超える一番小さい2のべき乗数を求める(ステップ2102)。今回の例では「256」である。

【0077】続いて、レコード長を仮に1、2、4、…、と2のべき乗数として以下の手順を行ない、それぞれの場合の平均書き込みオーバーヘッド時間を求める(ステップ2103)。仮のレコード長の上限は、前の手順で求めた、平均アクセス長を超える一番小さい2のべき乗数である。今回の例では、「256」である。

【0078】それぞれの仮レコード長について行なう計算は以下の通りである。まず、書き込み時に再暗号化処理の必要となる長さの平均、および1回のアクセスにつきレコード用暗号鍵と初期化データの計算が平均何回あるかを求める。再暗号化処理に必要な長さは、図18に示すような部分のことである。平均再暗号化処理長は、各アクセス長について「数1」に示す式で求めた値の平均値である。

【0079】

【数1】

$$(\text{再暗号化処理長}) = \begin{cases} R - L \% R & \cdots L \% R > 0 \text{ の時} \\ 0 & \cdots L \% R = 0 \text{ の時} \end{cases}$$

【0080】平均計算回数は、各アクセス位置とアクセス長の組について「数2」に示す式で求めた値の平均値である。なお、「数1」、「数2」の中で使用している記号の意味は図24に示す通りである。また、「L % R」とは、「L/Rの余り」のことである。

【0081】

$$(\text{計算回数}) = \frac{(P\%R) + L - 1}{R} + \begin{cases} 0 & \dots P\%R > 0 \text{ の時} \\ 1 & \dots P\%R = 0 \text{ の時} \end{cases}$$

【0082】次に、上のようにして求めた平均再暗号化処理長と平均計算回数、および暗号化処理速度、復号化処理速度、レコード用暗号鍵の生成にかかる時間、初期化データの生成にかかる時間から、「数1」、「数2」の計算に用いた仮レコード長を使った時の平均書き込み※
(平均書き込みオーバーヘッド時間) =

$$\frac{(\text{平均再暗号化処理長})}{(\text{復号化速度})} + \frac{(\text{平均再暗号化処理長})}{(\text{暗号化速度})} + \{(\text{レコード用暗号鍵生成時間}) + (\text{レコード用初期化データ生成時間})\} \times (\text{平均計算回数})$$

【0084】ここで、「数3」中の(平均再暗号化処理長) / (復号化速度)は、ファイルから読み込んだ再暗号化処理が必要な部分を復号化するのにかかる時間の平均である。また、(平均再暗号化処理長) / (暗号化速度)は、再暗号化処理が必要な部分を暗号化するのにかかる時間の平均である。また、{ … } × (平均計算回数)は、レコードごとに用意するレコード鍵と初期化データを計算するのにかかる時間の平均である。これら3つの式の和を求めると、書き込みオーバーヘッド時間の平均になる。

【0085】このようにしてそれぞれの仮レコード長について求めた平均書き込みオーバーヘッド時間のうち、その最少値を与える仮レコード長を選ぶ(ステップ2104)。これが書き込み時のオーバーヘッドを最少にするレコード長である。最後に、アプリケーションプログラムと選んだレコード長を組にしてレコードサイズ記憶部124に書き込む(ステップ2105)。

【0086】以上のように本実施形態においては、端末101のアプリケーションプログラム106からオペレーティングシステム108へ発行する記憶装置111へのデータ読み込み・書き出し要求を横取りし、書き出し要求の際にはデータを暗号化するための暗号鍵をネットワーク122で接続している鍵管理サーバ112から取り寄せ、この暗号鍵でデータを暗号化して記憶装置111に保存し、また、読み込み要求の際には記憶装置111から暗号化データを読み出し、この暗号化データを復号するための鍵を鍵管理サーバ112から取り寄せ、この暗号鍵で暗号化データを復号して取り出した元のデータをアプリケーションプログラム106に渡すようにしたため、暗号化や復号化の指示をユーザが明示的に行うことなく記憶装置111に保存するデータを自動的に暗号化し、読み出す時には自動的に復号することができ、煩わしい操作をユーザに強いることがないという効果が得られる。また、記憶装置111に保存するデータを自動的に保存するので、ユーザがデータの暗号化を忘

※オーバーヘッド時間は「数3」に示す式を用いて得ることができる。

【0083】

【数3】

れるという問題も起きない。

【0087】また、鍵管理サーバ112に対して暗号鍵を送ってもらう要求を出す前に、アプリケーションプログラム106を操作しているユーザに対してそのユーザを識別するための情報を入力させ、この情報の正しさをユーザが操作している端末101で確認した後にユーザの鍵を鍵管理サーバ112から取り寄せる、あるいはユーザが入力した識別情報を鍵管理サーバ112に送り、鍵管理サーバ112で受け取った識別情報の正しさを確認した後にそのユーザの暗号鍵を送り返すようにしているため、暗号化した機密データを保存してある携帯用の端末が盗まれたとしても、解読される危険性はこれらのデータが同一の記憶装置に保存されている場合に比べて小さくなり、データの機密性をさらに高めることができる。例えば、携帯用端末に暗号化データを保存しておき、社内でそのデータにアクセスしたい時は社内LANに接続し、社外からは社内ネットワークヘリモートアクセスして利用するようにしておけば、この携帯端末を不正に入手したユーザは、ユーザの識別情報を類推するだけでなく、リモートアクセス時のユーザ認証作業も行う必要があるので安全性がより高くなる。また、社内では携帯用端末を鍵管理サーバと同じネットワークに接続して重要データの編集や閲覧が可能のように構成でき、社外では鍵管理サーバが存在しないので、そのデータにアクセスできなくさせるように構成することができ、セキュリティを高めることができる。

【0088】さらに、アプリケーションプログラム106からオペレーティングシステム108へのデータ読み出し・書き込み要求を横取りするための仕組みは、オペレーティングシステム108が提供している場合には、既存のアプリケーションプログラムやオペレーティングシステムを書き換える必要がなく、しかも新たにハードウェアを置き換える必要もなく、最小限のコストでデータのセキュリティを高めることが可能である。

【0089】さらに、ICカードにユーザ固有の暗号鍵

を記録しておいて各ユーザ自身で暗号鍵を管理させる方法も考えられるが、ICカード自体の紛失によりデータの復号が不可能になったり、ICカードの読み書き機構を端末101に付加しておく必要があるので、コスト面で不利となる。しかし、本実施形態のように暗号鍵を鍵管理サーバ112から取り寄せるようにすることにより、ICカードの読み書き機構を備えていない端末であっても利用することができるという利点がある。

【0090】なお、本実施形態では、ファイルの暗号化に際し、ユーザ鍵をそのまま使わずにファイル暗号鍵をその都度生成しているが、ユーザ鍵でファイル自体を直接に暗号化してもよい。

【0091】また、ファイルを保存あるいは読み出すたびにユーザに対してユーザ名とパスワードを入力させているが、これらの情報を端末101のメモリに一時的に保存しておいて、次のファイル保存、読み出しの際にはこの一時保存してあるユーザ名とパスワードを使って鍵管理要求を生成するように構成することにより、ユーザの入力操作をさらに簡略化することができる。

【0092】さらに、取り寄せたユーザ鍵自体をメモリ上に一時保存し後で再利用する構成にすることにより、端末101と鍵管理サーバ112間の通信を行わずに済み、通信のオーバーヘッドを省いて処理スピードを速めることができ、ネットワーク122のトラフィックを減らすこともできる。

【0093】また、予めユーザがファイル保存の際には自動的に暗号化して欲しいファイルやディレクトリを指定するように構成することで、指定したファイルあるいは指定したディレクトリ以下にあるファイルだけに対して、暗号化や復号化を行うことができる。これにより、暗号化して保存する場所としてフロッピーディスクのような記憶媒体を表わすディレクトリも指定できるので、重要な情報をコピーされて持ち出されることを防ぐこともできる。

【0094】さらに、端末101および鍵管理サーバ112において実施形態で説明したような機能を実現する各機能部分は、具体的には同等機能を実現する処理プログラムで構成されるものであり、これらの処理プログラムはCDROMなどの記憶媒体に格納して、あるいはインターネット等の通信媒体を通じて利用者に提供することができるものである。

【0095】また、本実施形態では、レコードごとに暗号鍵と初期化データを計算していたが、読み書き時の速度を高速化するためにファイル暗号鍵とファイル初期化データをそのまま使ってもよい。この場合、本実施形態に比べて暗号化ファイルの安全性が少し失われることになる。また、レコード長の計算方法も変える必要がある。

【0096】レコード長を決めるためにアプリケーションのアクセス長だけでなく、オペレーティングシステム

によるファイルアクセスへのキャッシュの単位や、記憶装置上でのデータの記録単位（例えばセクタ長）を使うこともできる。

【0097】また、レコード長を求める時に、読み書きのアクセス記録の両方を使って平均書き込み時間を求めたが、厳密に読み書きを区別して計算してもよい。このためには、アクセス情報記憶部に読み書きを区別する欄を設ければよい。

【0098】また、レコード長はアプリケーションプログラム用のレコード長が求まってない時だけでなく、例えば一週間に一回というように定期的に計算してもよい。

【0099】また、レコード長を求めた結果、その値がそれまで使っていたレコード長と異なる値になった場合、既存のファイルを暗号化し直せばよい。

【0100】また、暗号化ファイルにレコード長を記憶しておく領域を設けておくことで、そのファイルを暗号化する際のレコード長を保存して置くことができる。このようにすれば、アプリケーションプログラムに最適なレコード長が変わった時に、既存の暗号化ファイルを新しいレコード長の値を使って簡単に暗号化し直すことが可能である。

【0101】

【発明の効果】以上のように本発明によれば、暗号化や復号化の指示をユーザが明示的に行うことなしに記憶装置に保存するデータを自動的に暗号化し、読み出す時には自動的に復号することができ、煩わしい操作をユーザに強いることがないという効果が得られる。また、記憶装置に保存するデータを自動的に保存するので、ユーザがデータの暗号化を忘れるという問題も起きない。

【0102】また、鍵管理コンピュータからユーザ別の暗号鍵を取り寄せるようにしているため、暗号化した機密データを保存してある携帯用の端末が盗まれたとしても、解読される危険性はこれらのデータが同一の記憶装置に保存されている場合に比べて小さくなり、データの機密性をさらに高めることができる。

【0103】また、ICカードにユーザ固有の暗号鍵を記録しておいて各ユーザ自身で暗号鍵を管理させる方法に比べ、ICカードの読み書き機構を備えていない端末であっても利用することができるという効果がある。

【0104】また、自動的に暗号化・復号化する際に問題となる、データ上書き時の再暗号化処理や初期化データ生成処理のオーバーヘッドも最小化できるので、このような最適化を行わない場合に比べてデータ上書き処理を高速化できる。

【図面の簡単な説明】

【図1】本発明を適用したコンピュータシステムの実施形態を示すシステム構成図である。

【図2】ユーザに配布する鍵情報レコードの構成図である。

【図 3】ユーザ配布情報作成手順を示すフローチャートである。

【図 4】認証トークンの構成図である。

【図 5】ファイル暗号化手順を示すフローチャートである。

【図 6】端末と鍵管理サーバとの間で送受されるメッセージのフォーマット構成図である。

【図 7】ユーザ鍵要求メッセージデータ部に含まれる元の内容を示す図である。

【図 8】ユーザ鍵要求メッセージの構成図である。

【図 9】ユーザ鍵返答メッセージデータ部に含まれる元の内容（ユーザ鍵あり）を示す図である。

【図 10】ユーザ鍵返答メッセージデータ部に含まれる元の内容（ユーザ鍵なし）を示す図である。

【図 11】記憶装置に記憶される暗号化ファイルの構成図である。

【図 12】ファイル復号化手順を示すフローチャートである。

【図 13】鍵管理アプリケーションの処理手順を示すフローチャートである。

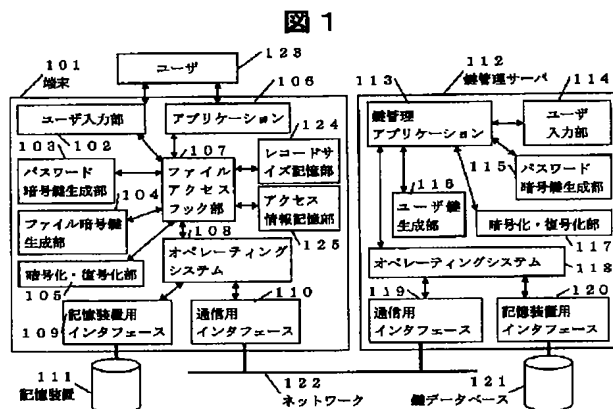
【図 14】暗号化の詳細手順を示すフローチャートである。

【図 15】アプリケーションプログラムからの書き込み要求の構成図である。

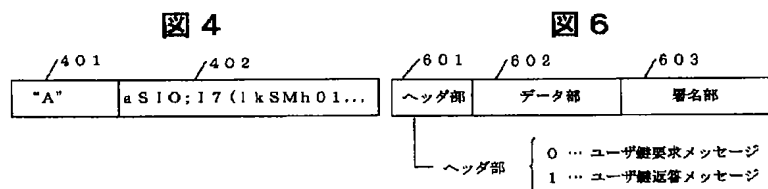
【図 16】レコードサイズ記憶部を表す図である。

【図 17】アクセス情報記憶部を表す図である。 *

【図 1】



【図 4】



【図 6】

図 6

* 【図 18】書き込み要求の実行によってファイル上に書き込まれる範囲と再暗号化処理の必要となる範囲を表す図である。

【図 19】復号化の詳細手順を示すフローチャートである。

【図 20】アプリケーションプログラムからの読み込み要求の構成図である。

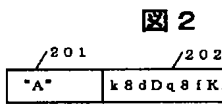
【図 21】レコード長を求める手順を示すフローチャートである。

【図 22】計算式に使われている記号の説明図である。

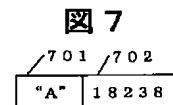
【符号の説明】

101…端末、102…ユーザ入力部、103…パスワード暗号鍵生成部、104…ファイル暗号鍵生成部、105…暗号化・復号化部、106…アプリケーション、107…ファイルアクセスフック部、108…オペレーティングシステム、109…記憶装置用インタフェース、110…通信用インタフェース、111…記憶装置、112…鍵管理サーバ、113…鍵管理アプリケーション、114…ユーザ入力部、115…パスワード暗号鍵生成部、116…ユーザ鍵生成部、117…暗号化・復号化部、118…オペレーティングシステム、119…通信用インタフェース、120…記憶装置用インタフェース、121…鍵データベース、122…ネットワーク、123…ユーザ、124…レコードサイズ記憶部、125…アクセス情報記憶部。

【図 2】

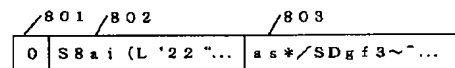


【図 7】



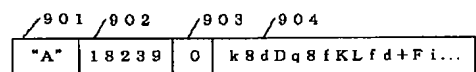
【図 8】

図 8



【図 9】

図 9



【図3】

【図10】

【図15】

図3

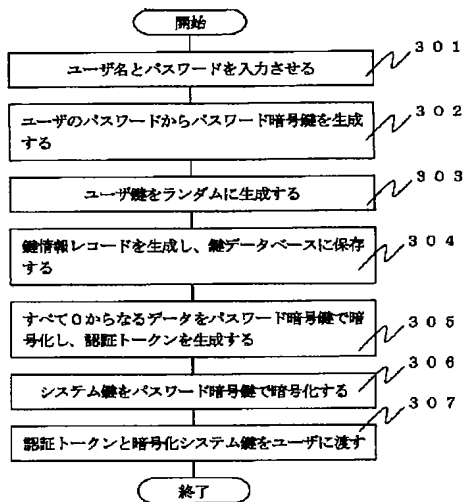


図10

1001	1002	1003
"A"	18239	1

図15

1501	1502	1503
1800	1000	2月3,843,0003,788,550...

【図16】

【図20】

図16

図20

1601	1602
表計算プログラム	256
ワープロプログラム	512
⋮	⋮
デフォルト	512

2001	2002
1024	150

【図17】

図17

1701	1702
アクセス開始位置	アクセスデータ長
256	512
0	128
⋮	⋮
1800	1000

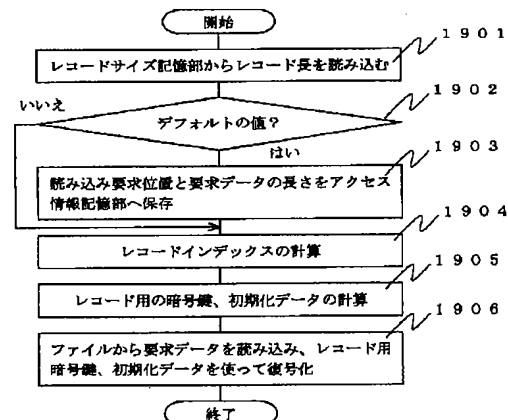
【図11】

図11

1101	1102	1103	1104
Akd803...	2u1adk...	DkNf83dK9sds*<...	SaS8sl...

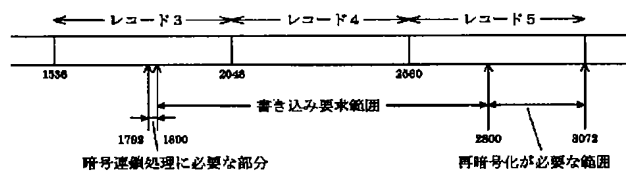
【図19】

図19



【図18】

図18



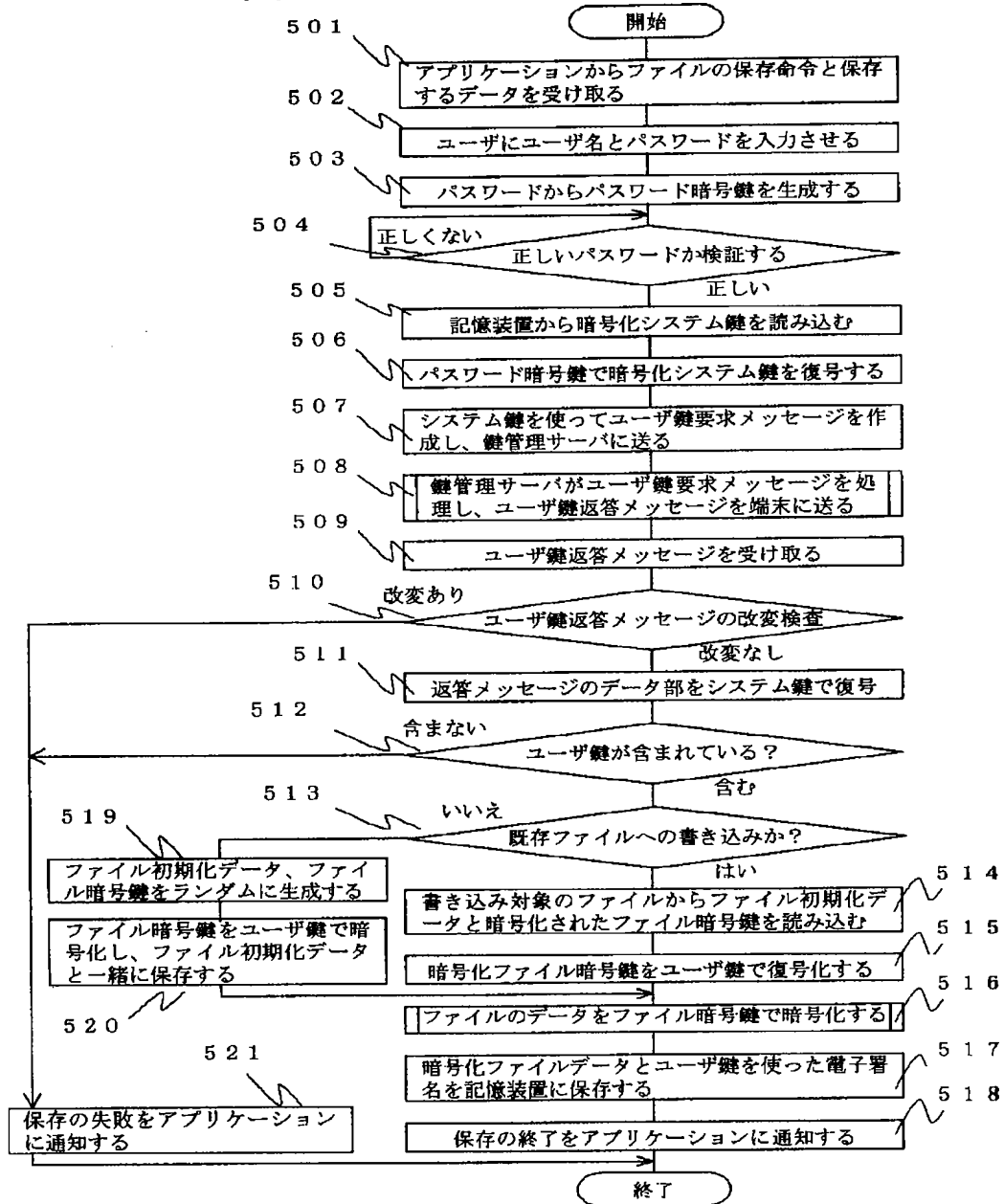
【図22】

図22

R	仮レコード長
P	アクセス開始位置
L	アクセスデータ長

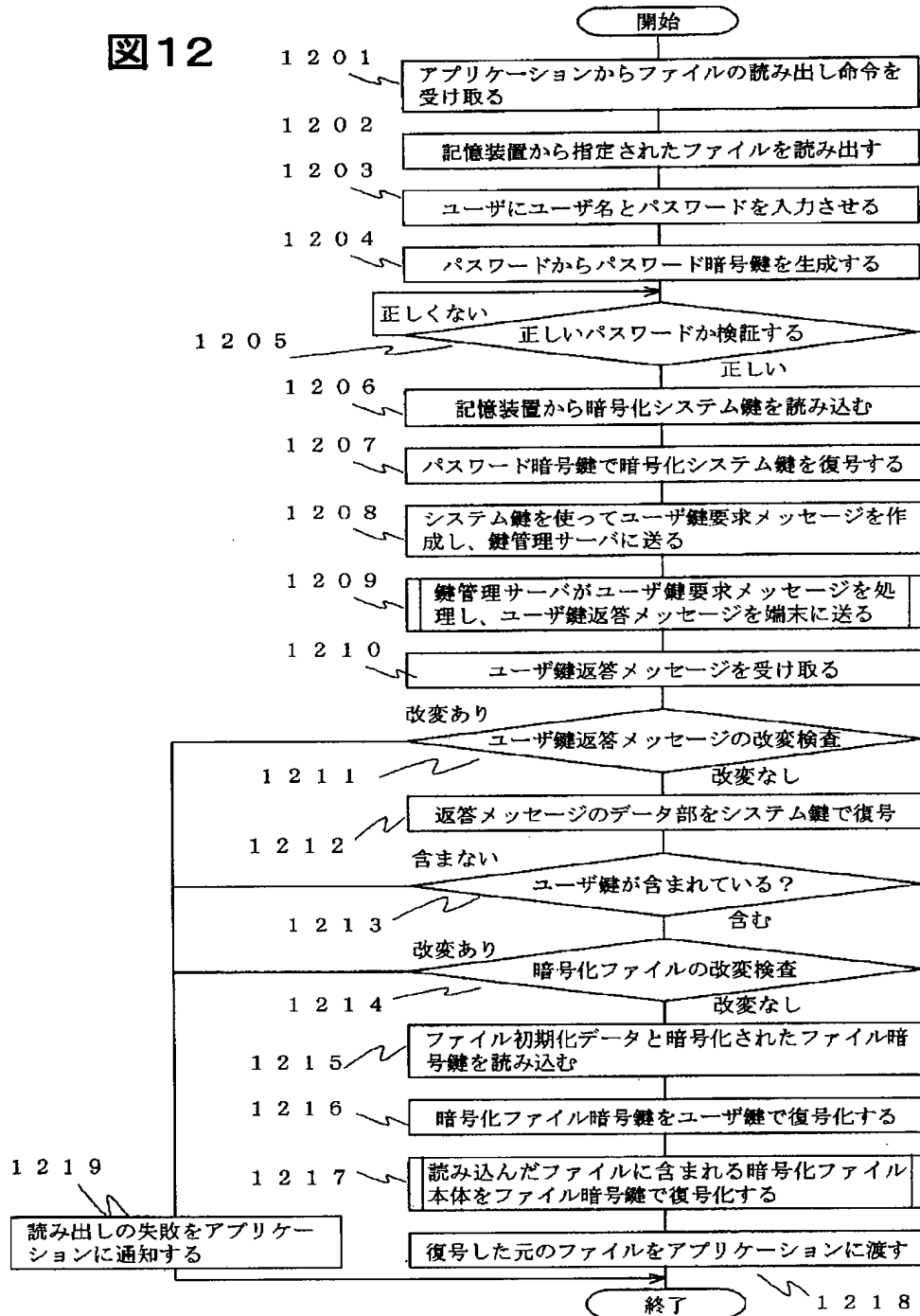
【図5】

図 5



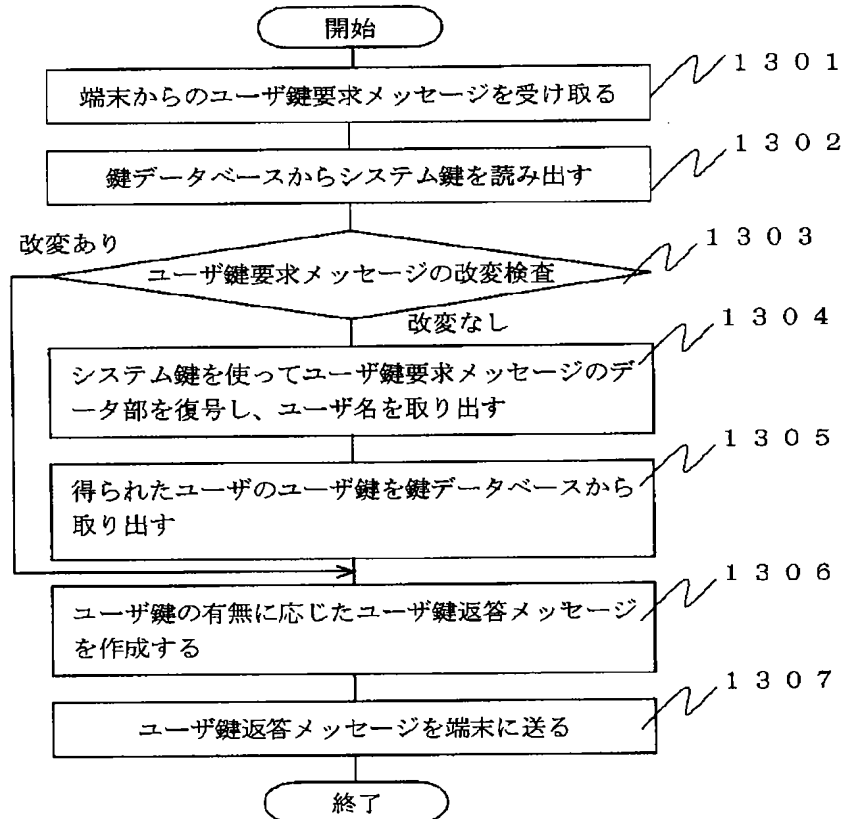
【図 1 2】

図 12



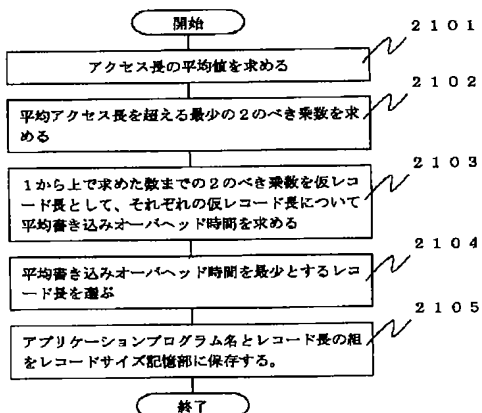
【図13】

図 1 3



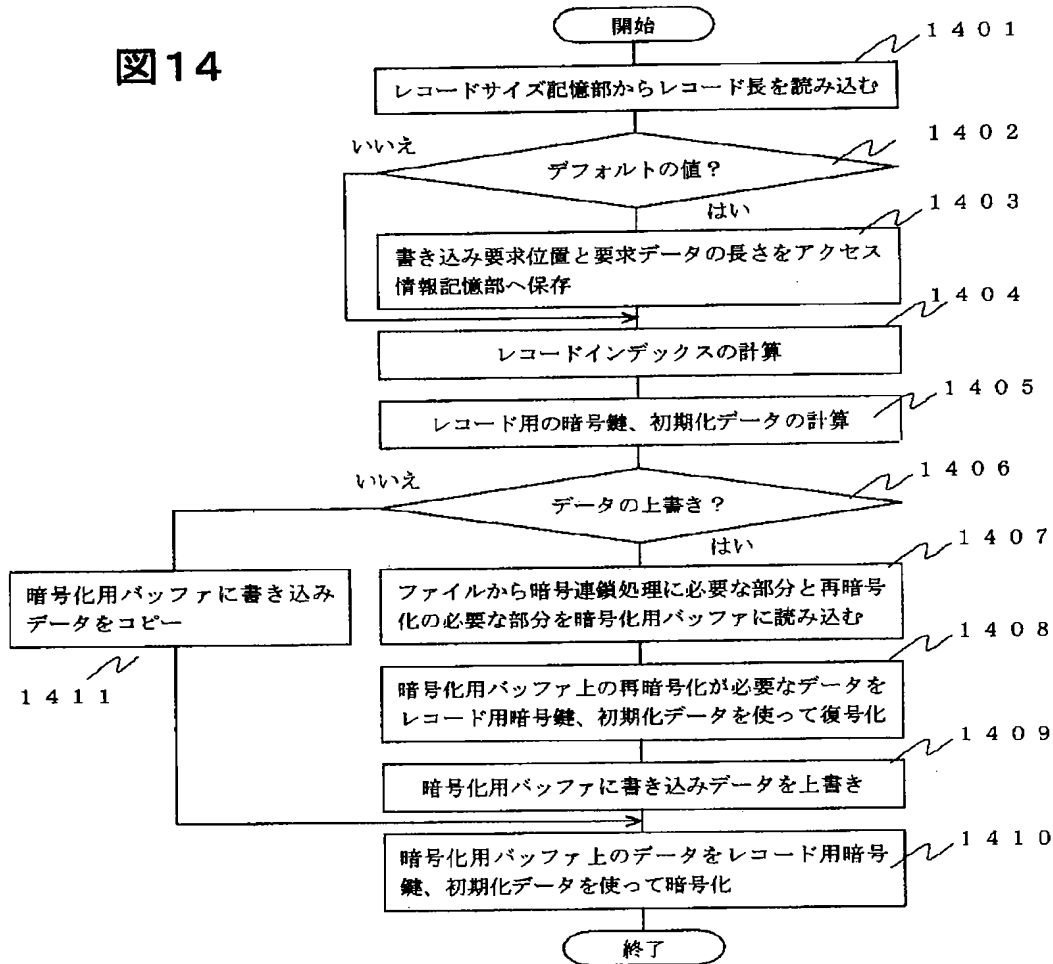
【図21】

図 21



【図14】

図14



【手続補正書】

【提出日】平成11年1月21日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】0035

【補正方法】変更

【補正内容】

【0035】続いて保存要求があったファイルを暗号化して保存するために、以下のようにして管理サーバ112からユーザ鍵を取り寄せる。まず、アプリケーションプログラム106を操作していたユーザAのユーザ名とパスワードをユーザ入力部102からユーザAに入力させる（ステップ502）。次に、入力されたパスワードからパスワード暗号鍵生成部103でパスワード暗号鍵を生成する（ステップ503）。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0052

【補正方法】変更

【補正内容】

【0052】続いて、暗号化されているファイル本体103からアプリケーションプログラム106の要求に対応した暗号化データを読み込み、復号して得たファイル暗号鍵を使って復号する（ステップ1217）。なお、復号の手順の詳細は図19で説明する。

【手続補正3】

【補正対象書類名】明細書

【補正対象項目名】0057

【補正方法】変更

【補正内容】

【0057】ユーザ鍵を得た場合には、図9に示したよ

うにユーザ名 9 0 1、レスポンスデータ 9 0 2、ユーザ鍵を含んでいることを示す値「0」9 0 3、ユーザ鍵 9 0 4 から成るデータを暗号化・復号化部 1 1 7 上でシステム鍵を使い暗号化したものを、ユーザ鍵返答メッセージのデータ部に含める。ユーザ鍵を得ることができなかった場合には、図 1 0 に示したようにユーザ名 1 0 0 1、レスポンスデータ 1 0 0 2、ユーザ鍵を含んでいないことを示す値「1」から成るデータを暗号化・復号化部 1 1 7 上でシステム鍵を使い暗号化したものを、ユーザ鍵返答メッセージのデータ部に含める。ユーザ鍵返答メッセージの署名部には、ユーザ鍵返答メッセージであることを示す値「1」を持つヘッダ部と、上記のように作成したデータ部からなるデータを、システム鍵を使って作成した署名を含める。

【手続補正 4】

【補正対象書類名】明細書

【補正対象項目名】0 0 6 2

【補正方法】変更

【補正内容】

【0 0 6 2】レコードサイズ記憶部 1 2 4 は、図 1 6 に示すように、表計算プログラムやワープロプログラムなどの各種のアプリケーションプログラム 1 6 0 1 ほどのレコードサイズ 1 6 0 2 を保存してある。もし、対応するレコード長が記録されていなかったら、レコードサイズ記憶部 1 2 4 の最後に記録されているデフォルトの長さを読み込む。ここでは、書き込み要求を行なったアプリケーションプログラムに対応する長さとして「5 1 2」を読み込むものとする。

【手続補正 5】

【補正対象書類名】明細書

【補正対象項目名】0 0 7 3

【補正方法】変更

【補正内容】

【0 0 7 3】図 1 9 において、まず、書き込み要求の時と同様に、読み込み要求を行なったアプリケーションプ

ログラムに対応するレコード長をレコードサイズ記憶部から読み込む（ステップ 1 9 0 1）。もし、対応するレコード長がなければデフォルトの値を読み込み（ステップ 1 9 0 2）、読み込み開始位置 2 0 0 1 とその長さ 2 0 0 2 をこのアプリケーションプログラム用のアクセス情報記憶部 1 2 5 に記録する（ステップ 1 9 0 3）。レコードインデックスの計算とレコード用の暗号鍵、初期化データの計算も、書き込み要求の場合と同様に行なう（ステップ 1 9 0 4、1 9 0 5）。

【手続補正 6】

【補正対象書類名】明細書

【補正対象項目名】0 0 8 0

【補正方法】変更

【補正内容】

【0 0 8 0】平均計算回数は、各アクセス位置とアクセス長の組について「数 2」に示す式で求めた値の平均値である。なお、「数 1」、「数 2」の中で使用している記号の意味は図 2 2 に示す通りである。また、「L % R」とは、「L / R の余り」のことである。

【手続補正 7】

【補正対象書類名】明細書

【補正対象項目名】0 0 9 8

【補正方法】変更

【補正内容】

【0 0 9 8】また、レコード長はアプリケーションプログラム用のレコード長が求まっていない時だけでなく、例えば一週間に一回というように定期的に計算してもよい。

【手続補正 8】

【補正対象書類名】図面

【補正対象項目名】図 1 2

【補正方法】変更

【補正内容】

【図 1 2】

図12

